Task 3

Effects of Augmentation

Alexander Garcia

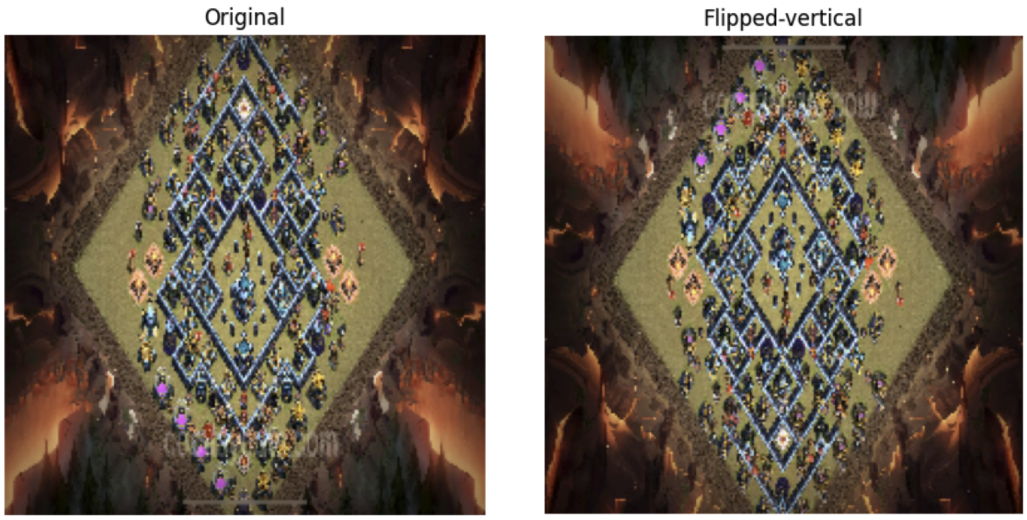
May 2025

Contents

1 Introduction

The goal of this phase is to test the effects of augmentation on the Clash of Clans data set. Augmentation applies different types of transformation to images such as rotating, flipping vertically or horizontally, zooming, and more. This helps create more training data for the model to learn from. While I expected a positive outcome from augmenting given my smaller (1042 images, 8 classes) data set, I experienced some overfitting when trying any augmenting except for vertical and horizontal flips.

2 Examples of Augmenting



(a) Base with no augmentation (b) Vertical flip augmentation

Figure 1: Augmentation examples

Augmenting has many options, and depending on the data set, some options may be more useful than others. A base at any level can be designed any way the user sees fit, so some augmentation seemed like it could be beneficial and represent a real possible base configuration. For example, mirror image bases actually do exist which would be similar to a vertical or horizontal flip. For my sample, I tried two different variations of augmenting, vertical and horizontal flips as well as some mild zooming and shift of width and height range. Too much zooming has a chance of removing the most important building from an image, especially if it is positioned in a corner, so it must be done carefully.

2.1 Effects of Augmenting

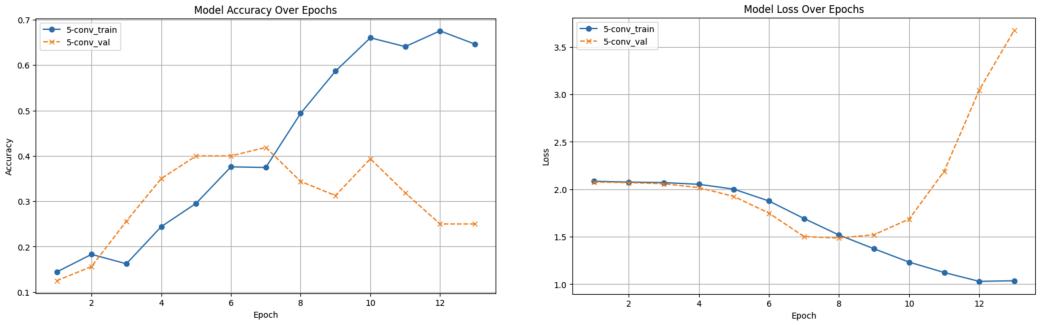
Augmentation has no effect on parameter count as seen in the image above. The model used is one of the final models from the previous tasks with 5 Conv layers, MaxPool (for downsampling), and then GlobalAveragePooling with a Dense layer of 200 neurons before the final 8 neuron softmax layer. I noticed mixed results when training. Below is a run when I tried to do slight

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 150, 150, 3)	0
conv2d_10 (Conv2D)	(None, 150, 150, 32)	896
max_pooling2d_8 (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_11 (Conv2D)	(None, 75, 75, 64)	18,496
max_pooling2d_9 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_12 (Conv2D)	(None, 37, 37, 128)	73,856
max_pooling2d_10 (MaxPooling2D)	(None, 18, 18, 128)	0
conv2d_13 (Conv2D)	(None, 18, 18, 256)	295,168
max_pooling2d_11 (MaxPooling2D)	(None, 9, 9, 256)	0
conv2d_14 (Conv2D)	(None, 9, 9, 512)	1,180,160
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 512)	0
dense_4 (Dense)	(None, 200)	102,600
dense_5 (Dense)	(None, 8)	1,608

Total params: 1,672,784 (6.38 MB)
Trainable params: 1,672,784 (6.38 MB)
Non-trainable params: 0 (0.00 B)

Figure 2: Enter Caption

width and height shifts as well as minor zoom (since I can't take chances on altering the town hall) which resulted in a very quick overfit.

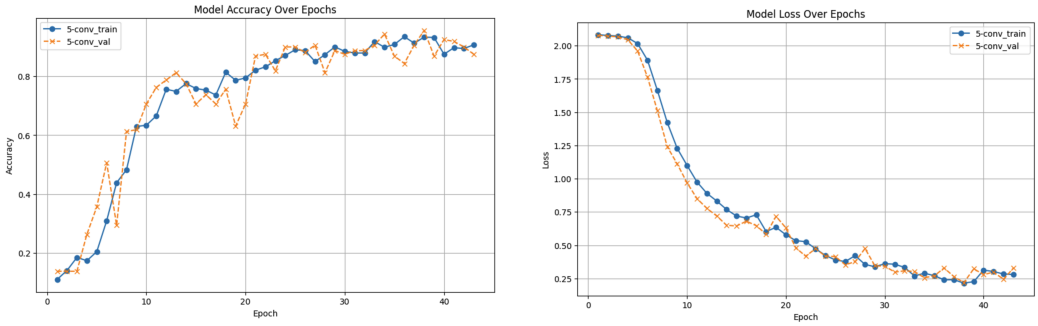


(a) Accuracy curve with augmented train

(b) Loss curve with augmented train

Figure 3: Metric curves of augmented train set

However, if I only used vertical and horizontal flips, the overfitting did not occur and the model performed similarly to itself when no augmentation was applied. Below are the accuracy and loss curves when only applying vertical and horizontal flips. The curves still spike during training on both the loss and accuracy, with a few spikes being quite drastic. The training time is similar to previous attempts at around 1 minute per epoch as well except the first epoch which ran for 593 seconds.



(a) Accuracy curve (b) Loss curve

Figure 4: Curves of vertical and horizontal flip augmentation

3 Conclusion

In summary, I attempted to add more training data through augmentation and analyze its effects. Overall, I tried a couple different augmentations like vertical and horizontal flipping, width and height shifts and zooming. However, these methods had different effects such as overfitting when too much augmentation was done and then similar performance to a non-augmented model. Overall, in the remaining tasks I chose not to include augmentation due to its lack of increasing accuracy or smoothing out the training spikes.